

**Boston University**

**Electrical & Computer Engineering**

EC464 Senior Design Project

**Final Project Test Report**

*Hybrid Battery Pack for Self-Launching eGliders*



Team #23

Team Members

Spencer P. Piligian, [spiligia@bu.edu](mailto:spiligia@bu.edu)

Daniel Kao, [dkao1234@bu.edu](mailto:dkao1234@bu.edu)

Yusheng Chen, [chenyush@bu.edu](mailto:chenyush@bu.edu)

Stephen Wang, [swang3@bu.edu](mailto:swang3@bu.edu)

Ismaeel AlAlawi, [ialalawi@bu.edu](mailto:ialalawi@bu.edu)

**I. Required Materials**

**Required Materials 1 (Manual Current Scheduling)**

*Hardware:*

* 3 x LithiumWerks A123 P6 (3.3V, 2500 mAh) lithium iron phosphate (LiFePO4) rechargeable battery cells
* 2 x Molicel P42A (3.6V, 4200 mAh) lithium-ion (Li-ion) rechargeable battery cells
* 3 x BeilaMoo 26650 3.7V single slot battery holders
* Surface mount technology (SMT) KeyStone Electronics 21700/20700 dual battery holder w/ coil springs
* IXYS VUO 190-08 NO7 three-phase diode bridge rectifier
* 2 x 1.5kW 1Ω resistors
* 40A automotive fuse
* TE Connectivity Potter & Brumfield T9GS1L14-12 30A relay
* TeamTriforceUK A50S V2.1 (50A burst) VESC-supported motor controller
* XT30 connector (battery input)
* MR30 connector and cable cover (motor output)
* 2 x pico-clasp connector shells
* microUSB to USB cable
* 10 µH inductor
* Male-to-male jumper wires
* Test Products Intl. (TPI) dual temperature thermocouple digital thermometer 343
* 1 x 10kΩ resistor
* Alligator clips
* Crocodile clips
* Hook clips
* Banana plugs
* Electrical ring tongue terminals AWG 18-22
* 2 x TE Connectivity TE2500B1R0J 50 W - 2500 W chassis-mount power resistor
* AWG 8, 12 & 18 wires
* ACDelco SuperALKALINE 6LR61 9V battery
* PKCELL Extra heavy duty 6F22 9V battery
* 2 x battery snap connectors (9V)
* Electrical insulation tape and masking tape
* 5 x MCR Safety safety glasses
* 2 x Multimeter

*Software:*

* VESC Tool (free version) GUI developed by Benjamin Vedder
  + Qt Toolkit
  + .zip file extractor (Windows, Linux, Mac OS X, and Android supported)
  + Source code: <https://github.com/vedderb/vesc_tool>

**Required Materials 2 (Automated Current Scheduling)**

*Hardware:*

* TeamTriforceUK A50S V2.1 (50A burst) VESC-supported motor controller
* XT30 connector (battery input)
* MR30 connector and cable cover (motor output)
* 2 x pico-clasp connector shells
* microUSB to USB cable
* Male-to-male jumper wires
* Test Products Intl. (TPI) dual temperature thermocouple digital thermometer 343
* 0-30V/0-5A lab power supply (OWON P4305)
* Electrical insulation tape and masking tape
* 5 x MCR Safety safety glasses
* Multimeter

*Software:*

* VESC Tool (free version) GUI developed by Benjamin Vedder
  + Qt Toolkit
  + .zip file extractor (Windows, Linux, Mac OS X, and Android supported)
  + Source code: <https://github.com/vedderb/vesc_tool>
* VESC-Lisp, and LispBM Scripts
  + ANSI Common Lisp GNU v2.49 (2010-07-07 build) compiler (REPL)
  + For Mac OS X, install via MacPorts
  + For Windows and Linux (Ubuntu, Debian, Sun Solaris, etc.), install directly from <https://clisp.sourceforge.io/>
  + LispBM (Lisp Black Magic dialect) documentation by Joel Svensson: <https://github.com/svenssonjoel/lispBM/blob/master/doc/lbmref.md#eval>

**II. Setup**

**Setup 1 (Manual Current Scheduling)**

In order to properly conduct this in-lab prototype test, there are several hardware and software requirements that need to be met. On the hardware side, the battery circuit includes a 9V DC power supply that is connected to a 30A relay and that relay is connected to a parallel connection of LiFePO4 and Li-ion batteries in a 3:2 ratio respectively, in which a Schottky barrier diode is placed in series with the Li-ion batteries to prevent reverse leakage current from the LiFePO4 cells to the Li-ion cells. It is important to note that this Schottky barrier is part of a diode bridge rectifier (i.e. terminals are adjusted accordingly in order to retrieve a single Schottky barrier diode from the rectifier circuit). There exists a 40A fuse that is connected in series with the motor controller and the current sensor for safety standard purposes. In the event of a full short we would expect a current spike up to 550A – ~9.9V/18mΩ (internal resistance) – causing the fuse to blow nearly instantly. The motor controller is properly connected to an inductor using connection shells and the inductor is then coupled with two power resistors in order to dump energy – by connecting the inductor in series with the power resistor, the current can be tuned easily through the motor controller. For safety reasons, we will be using a thermocouple that will log the temperature of the batteries; the digital thermocouple of the motor controller will log the temperature of the MOSFETs in the motor controller. The battery and motor controller voltages will also be recorded appropriately using multimeters. On the software side, using VESC Tool software, the motor and battery current readings are plotted over time graphically in order to determine the amount of current flowing through the motor that is required for the battery to reach a discharge current of 15A.

**Setup 2 (Automated Current Scheduling)**

For the second portion of testing we will be testing automated power scheduling software through the use of a power supply as a safe proof of concept. For this we will need a TeamTriforceUK A50S V2.1 (50A burst) VESC-supported motor controller connected to a lab power supply which is controlling a three phase fan motor. In this test, after having connected the power supply and fan motor, we set the voltage to 18V which is what the motor requires and set the maximum supported current rating of 5A accordingly. Then, in the VESC Tool software, we configure the duty cycle to 0.10 to simply get the motor to start rotating, and then adjust the current rating manually initially to 3A, the lowest supported current. Then, the lispBM *battery\_demo.lisp* is run to automatically set the motor current transitions. At first, during the “step-down testing phase”, the current rating is set to a high current of 4A for 10 seconds, then transitions to a medium current of 3.5A for 20 seconds, and lastly, transitions to a low current of 3A for 30 seconds. Afterwards, during the “step-up testing phase”, the current rating is stepped up back to 3.5A for 20 seconds, and stepped up further to 4A for 10 seconds, until the demo ends. In conjunction with the listpBM script running, the motor current, battery current, MOSFET temperature, and motor RPM are tracked in real-time using the RT Data Analysis plug-in of the VESC Tool.

**III. Pre-testing Setup Procedure**

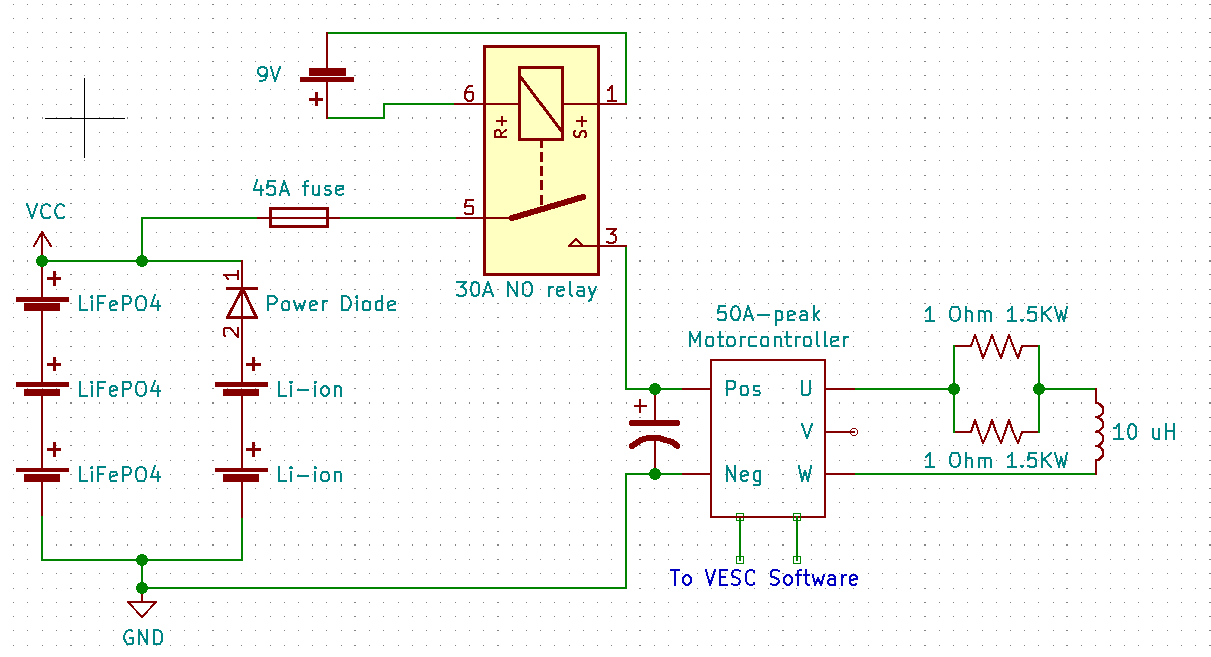
**Pre-testing 1 (Manual Current Scheduling)**

* *Battery testing circuit hardware side (see* ***Figure 1*** *for visual reference)*
  + Attach 9V connector battery to the relay (top left and right connections) by soldering battery snap connectors to energize the relay’s coil
  + Place each of the three LiFePO4 batteries in their respective single slot battery holders
  + Place both Li-ion batteries in the dual slot battery holder
  + Solder the wires connecting the LiFePO4 battery holder contacts in series and solder the first ring terminal wire to one of the LiFePO4 battery holders accordingly and the other to the Li-ion dual battery holder to create the parallel connection
  + Connect the ring terminal connected to the LiFePO4 battery holders into terminal ~C (phase-C) of the rectifier and the other terminal (which is connected to Li-ion batteries) to terminal B- (see **Figure 3** for rectifier reference schematic)
  + Solder the LiFePO4 battery holder and rectifier junction into the 30A relay’s bottom right terminal
  + Solder the 40A fuse into the bottom-right terminal of the Li-ion battery holder and connect its other end to the negative top terminal of the current sensor
  + Connect the positive top terminal of the current sensor to the motor controller positive XT30 connection; the negative terminal of this connection should be connected to the multimeter accordingly for voltage readings.
  + Connect the positive terminal of the motor controller MR30 connector into the positive terminal of the multimeter to complete the connection for motor controller voltage readings
  + Connect the A50S motor controller to the laptop using micoUSB-USB cable
  + Connect the positive and negative terminals of the motor controller-motor junction to the positive and negative terminals of the power resistor respectively
* *Diode, 9V batteries, and properly soldered electrical connection testing (if necessary)*
  + Using a multimeter, test the voltages/currents across the terminals of the Schottky junction diode and the 9V batteries, as well as other electrical connections in the circuit to ensure that the connections aren’t faulty, i.e. improperly soldered or connected.
* *Electrical insulation* *of electrical contacts*
  + Ensure that any exposed metallic wire ends are properly insulated for safety purposes
  + Coat sufficient amount of tape – typically used black electrical insulating tape – fully around metallic wire ends
  + Confirm that insulating tape does not disrupt the functioning of other electrical components for testing.

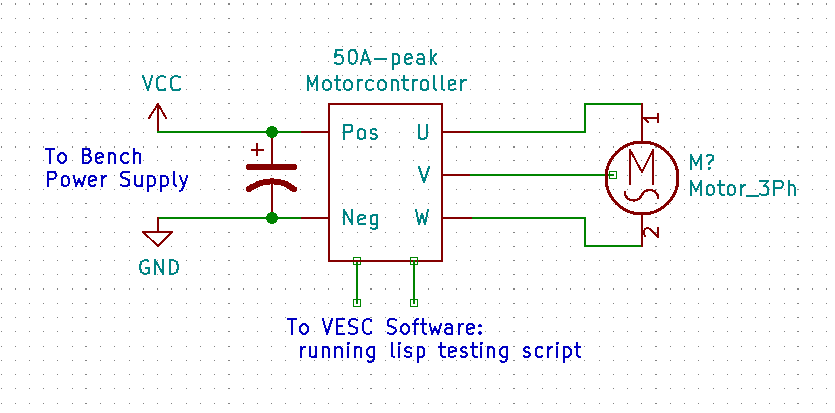
**Pre-testing 2 (Automated Current Scheduling)**

* *VESC Tool setup software side*
  + Create an account on <https://vesc-project.com/> > click on “VESC tool” tab > purchase VESC Tool Free for €0.00 (no billing information required) > click on “Purchased Files” link (top right corner) > download and extract “vesc\_tool\_BETA\_ALL.zip” (or for Windows users, download “vesc\_tool\_free\_windows.zip”)
  + Alternatively install directly via source code: <https://github.com/vedderb/vesc_tool>
  + For Windows and Linux users, Install REPL Lisp compiler via SourceForge.
  + For Mac OS X users install MacPorts v2.7.1
    - Install Xcode and Xcode Command Line Tools
    - Agree to Xcode license in Terminal: sudo xcodebuild -license
    - Install MacPorts according to version of OS X (convenient installation through .pkg installer for version 10.15+) on <https://www.macports.org/install.php>
    - After MacPorts installation, type sudo port install clisp to install common lisp implementation (interpreter, compiler, debugger, etc.).
  + Once VESC Tool is installed, click on the “Connection” option on the left-hand side of the interface
    - Select “cu.Bluetooth-Incoming-Port” as the serial port, and set the default Baud rate to “115200 bps”
    - Click on “Refresh serial port list” after physically connecting the motor controller to the computer
    - Click on “Connect” and under CAN-devices, the A50S motor controller name should show up on the bottom left – if this is not the case, close the VESC tool, and repeat the aforementioned steps
* *Electrical insulation* *of electrical contacts*
  + Ensure that any exposed metallic wire ends are properly insulated for safety purposes
  + Coat sufficient amount of tape – typically used black electrical insulating tape – fully around metallic wire ends
  + Confirm that insulating tape does not disrupt the functioning of other electrical components for testing.

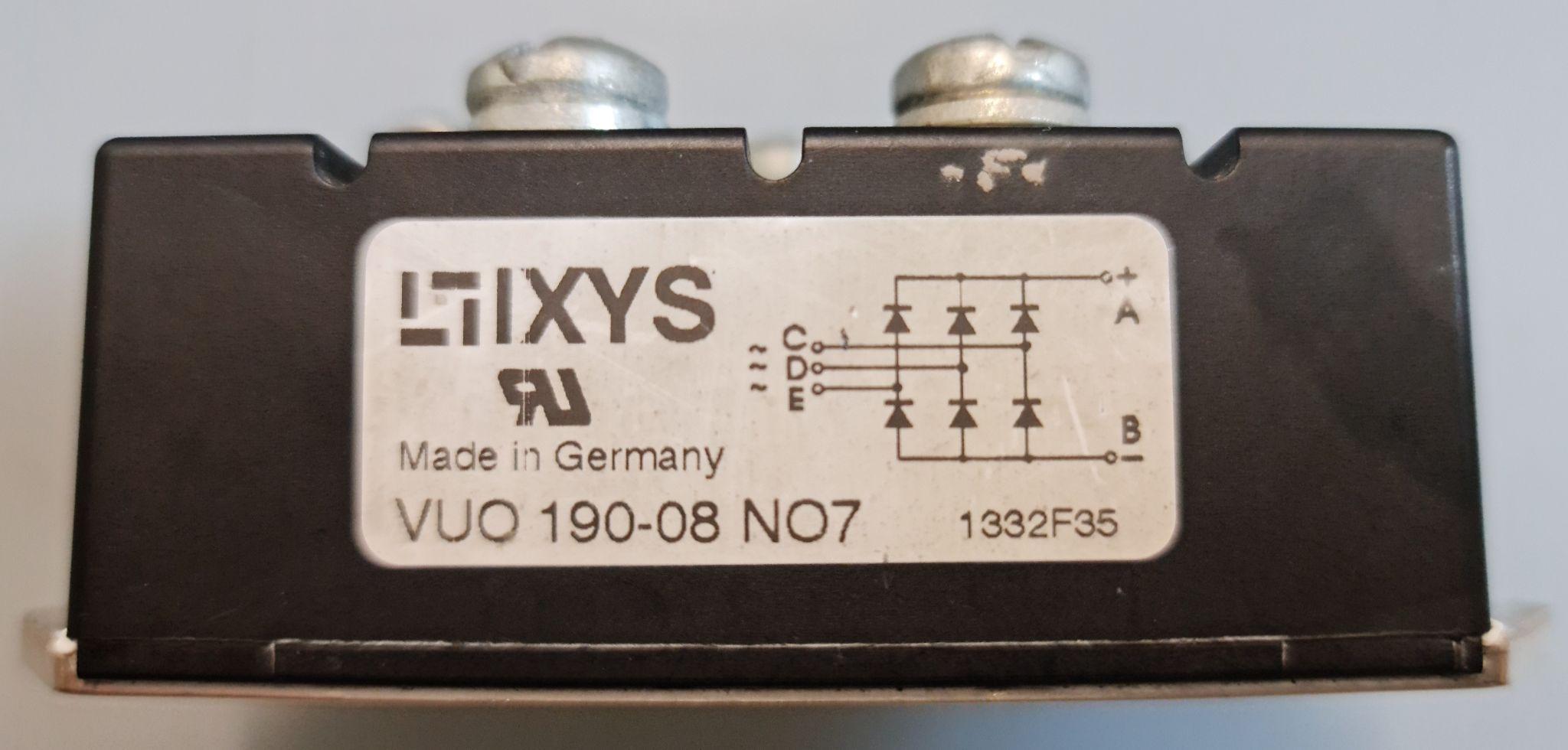
**Figures:**

****

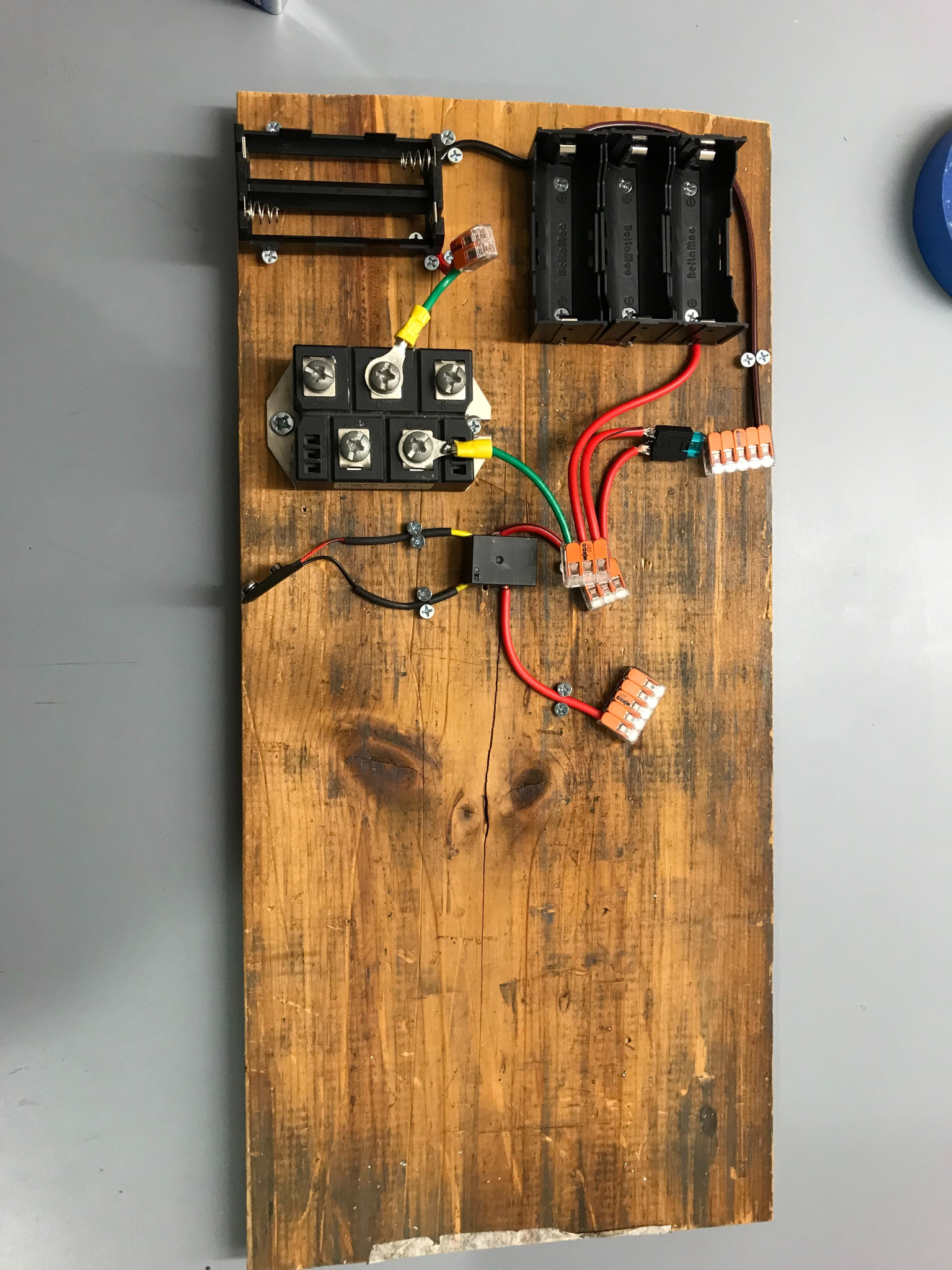
**Figure 1:** KiCad EDA design schematic of In-lab test 1 (manual current scheduling). Schematic in terms of motor controller design[[1]](#footnote-0).

****

**Figure 2:** KiCad EDA design schematic of In-lab test 2 (automated current scheduling). Schematic in terms of motor controller design .

****

**Figure 3:** Reference schematic of three-phase diode bridge rectifier. Terminals ~C and B- are used to retrieve a single Schottky barrier from the rectifier.



**Figure 4:** Battery testbench reference

**IV. Testing Procedure**

**Testing Procedure 1 (Manual Current Scheduling)**

1. Start assembling the circuit based on Figure 1. Make sure you have all components available
2. The fuse relay and power resistors are connected in series to the power supply and parallel to the dual battery packs (Lithium-Ion and LiFePO4)
3. The thermistor is in contact with each battery for two seconds to get temperature readings. Then the motor controller is wired directly to the thermocouple digital thermometer for our separate temperature monitoring system accordingly - for alternate temperature sensing, use a thermal IR camera
4. Before running the test, setup the multimeter connection point to probe points as close to the motor speed controller as possible (this is due to the voltage drop across the wires at higher amps)
5. Then begin the test experiment by closing the relay in the circuit and manually adjusting *i\_motor* (motor current) at the bottom left corner of the VESC tool from 0 A, until the point the battery current *i\_batt* reaches 15A in conjunction. Note that VESC software handles the temperature logging of the MOSFETs on the motor controller, and this can be found on the top under the temperature tab
   1. Record the current readings of each battery pack circuit frequently and continue to monitor the temperature.

**Testing Procedure 2 (Automated Current Scheduling)**

1. Start assembling the circuit based on Figure 2. Make sure you have all components available and all connections are correct.
2. Select “VESC Dev Tools” on the bottom left side of the VESC Tool interface and click on the “Lisp” tab after having successfully connected the A50S motor controller as per the pre-testing procedure
3. Download *battery\_demo.lisp* and add it to your selected VESC tool lisp file path, and under the LispBM script editor, click on “Open” to open *battery\_demo.lisp*
4. Make sure that “AutoRun” is unchecked at the top to prevent the script from immediately running on its own after uploading it. After having done this, click on “Upload” to upload the script.
5. Ensure that *battery\_demo.lisp* is currently flagged as the “main” script running on the VESC Tool by checking that “(main)” is written after the file identifier on the lispBM script tab
6. The script *battery\_demo.lisp* automates motor current transitions by directly communicating with the VESC A50S motor controller in setting two phases for testing: step-down and step-up phases. For the step-down phase, which is the first phase, a high current of 4A for 10 seconds, then transitioning into the next testing phase of medium current of 3.5A for 20 seconds, and lastly a low current testing phase of 3A for 30 seconds. For the step-up phase, the current is stepped up back to 3.5A for 20 seconds and ends with 4A for 10 seconds
7. When ready, press the “Run” button to run the script, then under the window on the left-hand side, select RT (real-time data) and on top, select the current to display the current-time plots for the motor in conjunction with the running script. All of these parameters will be recorded in conjunction with the MOSFET temperature logging, motor RPM, power dissipated by the fan, the motor duty cycle, the amount of Ah drawn, and the amount of Wh drawn accordingly, as well as the motor input voltage.

**V. Measurable Criteria**

**Measurable Criteria 1 (Manual Current Scheduling)**

The criteria for successful testing of this prototype of the eGlider hybrid battery pack are as follows:

1. Diode preventing reverse saturation current from LiFePO4 batteries to Li-ion batteries
2. Temperature of batteries is at room temperature (~23°C-25°C), and does not overheat (≥ 60°C)
3. Current is being drawn from LiFePO4 batteries
   1. Current sensor displaying current drawn from LiFePO4 batteries
   2. Collect data on current across adjustable load (especially the current at the discharge current of 5A)
4. Log aforementioned currents graphically over time using VESC tool software, as well as voltage and other temperature data acquired
5. MOSFETs on A50S motor controller do not overheat during testing (< 70°C)

**Measurable Criteria 2 (Automated Current Scheduling)**

The criteria for successful testing of this prototype of the eGlider hybrid battery pack are as follows:

1. MOSFETs on A50S motor controller do not overheat during testing (< 70°C)
2. Successful step-down motor current testing – the motor current should be fixed at 4A in the beginning for 10 seconds, and transition smoothly to 3.5A for 20 seconds, and also transition smoothly down to 3A for 30 seconds, without abruptly stopping or tripping the power supply
3. Successful step-up motor current testing – the motor current should be fixed at 3A in the beginning, from the previous testing phase, and transition down smoothly to 3.5A for 20 seconds, in which it ends with 4A for 10 seconds, without abruptly stopping or tripping the power supply
4. The power dissipated by the fan should be below 100W
5. The duty cycle should be fixed for the given current rating test phase (e.g. 30%-40% duty cycle with minimal intermittence for 4A motor test)
6. The Ah drawn and Wh drawn should be fixed for the given current rating test phase
7. The motor voltage should be fixed at the supplied 18V

**VI. Score Sheet and Planned Data Collection**

**Score Sheet and Data Collection 1 (Manual Current Scheduling)**

| **Time Elapsed (s)** | **Battery Dischar-ge Current (A)** | **Li-ion Battery Voltage (V)** | **LiFePO4 Battery Voltage (V)** | **Battery Temperatures Li-ion (°C**) | **Battery Temperatures LiFePO4 (°C**) | **MOSFET**  **Temp (< 70°C)** |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | **0** | **9.28** | **9.85** | **23.8** | **21.8** | **Y** |
| 20 | **4** | **9.23** | **9.72** | **23.8** | **21.8** | **Y** |
| 40 | **8** | **9.18** | **9.3** | **24.1** | **21.7** | **Y** |
| 60 | **12** | **9.0** | **9.13** | **24.1** | **21.9** | **Y** |
| 80 | **9.8** | **9.12** | **9.15** | **24.1** | **22.8** | **Y** |
| 100 | **8** | **9.13** | **9.17** | **24.1** | **23.2** | **Y** |
| 120 | **8** | **9.14** | **9.61** | **24.1** | **23.2** | **Y** |
| 140 | **4** | **9.21** | **9.79** | **24.1** | **23.2** | **Y** |
| 160 | **0** | **9.04** | **9.81** | **24.1** | **23.2** | **Y** |
| 180 | **0** | **9.26** | **9.83** | **24.1** | **23.2** | **Y** |

**Score Sheet and Data Collection 2 (Automated Current Scheduling)**

| **Motor Current (A)** | **Time Elapsed (s)** | **Power**  **(W)** | **Ah**  **Draw**  **(mAh)** | **Wh**  **Draw**  **(Wh)** | **Motor Voltage**  **(V)** | **RPM** | **Duty Cycle**  **(%)** | **MOSFET**  **Temp (< 70°C)** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 10 | **25** | **51.2** | **0.92** | **17.9** | **21207** | **30.2** | **34.9** |
| 3.5 | 20 | **17** | **54.3** | **0.98** | **17.9** | **18340** | **31.6** | **35** |
| 3 | 30 | **11** | **59.8** | **1.08** | **18** | **16110** | **29.8** | **34.3** |
| 3.5 | 20 | **17.2** | **64** | **1.15** | **17.9** | **18753** | **33.9** | **34.7** |
| 4 | 10 | **25** | **71.5** | **1.29** | **17.9** | **22035** | **40.7** | **36.1** |

**IX. Measurements Taken**

The measurements taken during our tests are meant to show that our circuit is working correctly and all components are working properly.

**Test 1 (Manual Current Scheduling)**

1. **Li-ion Battery Voltage:** This is taken because we need to make sure that our diode is working properly.For our text the correct operation of the diode is expected to be reverse biased. We expected to see no voltage change in the Li-ion battery pack because it is providing no power to the motor controller. This is seen in our Score sheet above.
2. **LiFePO4 Battery Voltage:** This is taken because we need to know how many volts are going into the motor controller as well as the state of charge of our battery pack. Also ensures the LiFePO4 is the only battery pack providing power to the motor.
3. **Battery Temperatures Li-ion:** This is taken to ensure that we don’t overheat the Li-ion battery and they react to current as expected.
4. **Battery Temperatures LiFePO4:** This is taken because we need to ensure that we don’t overheat the LiFePO4 battery.
5. **Motor Voltage using Multimeter:** This is taken because we will need to see the voltage drop from our batteries to the motor controller.
6. **Battery Discharge Current:** This is taken because this can demonstrate how many amperes that we are pushing into the motor controller.
7. **Motor Discharge Current:** By adjusting the motor current in the VESC software we also adjust the current coming from the batteries.
8. **Motor Controller MOSFET Temperature:** This is taken because we need to ensure that we didn’t burn out the motor controller with such high ampere used.

**Test 2 (Automated Current Scheduling)**

1. **Power:** This is taken because we want to know how much power it can provide to the motor. This can help us to adjust the power schedule for the take-off.
2. **Ah Draw:** This is taken because we want to know how much ampere we are creating and so we can make adjustments for the ampere we are producing every second.
3. **Wh Draw:**This is taken because we want to know how much power we are creating and so we can make adjustments for the ampere we are producing every second.
4. **Motor Voltage:** This is taken because we will need to know how much voltage is supplied to the motor.
5. **RPM:** This is taken because we will need to know the speed of the motor. We need to adjust the rpm so that we can adjust the power schedule.
6. **Duty cycle:** This is taken because this allows us to know the peak current.
7. **Motor Controller MOSFET Temperature:** This is taken because we need to ensure that we didn’t burn out the motor controller with the given motor currents tested.

**X. Conclusion**

**Test 1 (Manual Current Scheduling) Conclusion**

Overall, our final project testing was successful in that we are capable of manually testing the control of our motor controller with different amperage while building upon our previous test goals. During the demo we measured both Lithium-ion and LiFePO4 battery voltages, current (battery discharge rate) and temperature to ensure that we are operating in a safe temperature range for our batteries. During testing we set a steady increment of 4 A to a peak of 12 A and mirror the same rate when decreasing the current. Based on our previous test plan we are now able to adjust the current of the motor to allow the battery to reach 12 A, maximum no more than 15 A. During setup the load we used was ~0.5Ω and a fixed voltage of ~10 V. We were unable to push the current further due to voltage limitation. Another way in which this test was different was the decrease in the in-series resistance of the load by connecting two 1Ω power resistors in parallel to halve the resistance and double our available current ceiling. We also used a 10 µH inductor instead of a motor in our manual test. This helped simulate a “small-signal model” in lieu of the original motor used.

**Test 2 (Automated Current Scheduling) Conclusion**

Based on the valuable lessons we learned in our previous test bench and feedback, we created a more robust prototype that enhances the safety of our design and test procedure while using automated test schedules. This test runs on a fan motor instead of the 10 µH inductor and resistors. It also uses a 5 A/18 V lab power supply instead of the batteries. For our testing procedure we ran the test for approximately two minutes to simulate eGlider initial launch and gradual current consumption for the rest of the ride based on the timeline in our data above. That is to say, in the beginning of the launch, the motor consumes a large amount of current supplied by the battery for a short period of time, and then eventually consumes smaller amounts of current over larger periods of time to sustain its altitude mid-air. In this testing, it was evident that it is certainly possible to automate the eGlider motor current testing by means of running the LispBM script via the VESC Tool software. It indicates that several relevant parameters can be traced and adjusted accordingly. From this test’s data, it can be concluded that the motor dissipated the expected amount of power with steady MOSFET temperatures, i.e. the MOSFETs did not overheat. The RPM rate of the motor remained approximately fixed for the given current ratings; albeit, it is important to note that there is somewhat of a discrepancy between the current ratings of approximately 0.1 A, which is mainly due to either the motor controller or the VESC Tool. The duty cycle also remained approximately the same, but spiked at the end as the current rating also spiked initially due to a software bug. In terms of the implications of this, the program can be further improved and expanded by connecting the motor controller to a VESC-readable BMS. This way, the team can use many of the VESC-commands that can help both set and get data to and from the BMS for more large-scale battery testing for the client. Although, our BMS doesn’t directly support VESC, which implies that we have to use a serial communication protocol in order to let the BMS interface with the motor controller, and this can be done through UART. As such, further work and testing should be conducted on UART communication in order to set up a more powerful automated testbench as part of our final package for the client at the end.

Overall, we are satisfied with our ability to meet our initial design requirement and customer expectations. Our next step is to fine tune our design prototype with some further testing, develop the battery case and battery display for temperature sensing, and combine it into a small scale prototype plan that our client can follow to build the final desired battery pack for the eGlider.

1. 45A fuse in schematic should be 40A in accordance with pre-testing 1. [↑](#footnote-ref-0)